IN THE CLAIMS

1. [cancelled]
2. [cancelled]
3. [cancelled]
4. [cancelled]
5. [cancelled]
6. [cancelled]
7. [cancelled]
8. [cancelled]
9. [cancelled]
10. [cancelled]
11. [cancelled]
12. [cancelled]
13. [cancelled]
14. [cancelled]
15. [cancelled]
16. [cancelled]
17. [cancelled]
18. [cancelled]
19. [cancelled]
20. [cancelled]
21. [cancelled]
22. [cancelled]
23. [cancelled]
24. [cancelled]

PATENT

25. [NEW] A computer-readable medium having stored thereon computer-readable instructions, which, when executed by a computer, cause said computer to perform the following process:

receive user input data that defines primitives of a conceptual model, said conceptual model being comprised of an object model, a dynamic model, a functional model and a presentation model which together define the complete functionality of a target computer program to be automatically generated and which defines complete interface mechanisms for interaction of user or other processes with the functionality of said target computer program, no other software code, code component, code libraries or any other third party software artifact being necessary to completely define the functionality of said target computer program and its interface mechanisms;

automatically converting said user input data into data structures in the form of formal language statements organized by a formal language syntax, the collection of all such formal language statements forming a formal language specification;

validating said formal language specification to ensure it is complete, correct and unambiguous to prepare said formal language specification for automatic translation into complete, operative code of said target computer program which is the functional equivalent of said conceptual model and having a user interface defined by the primitives entered by said user which define said presentation model.

26. [NEW] The computer-readable medium of claim 25 wherein said computer readable instructions control said computer to receive user input which defines one or more service presentation patterns, one or more instance presentation patterns, one or more class population presentation patterns and one or more master-detail presentation

patterns as part of said interface mechanisms

27. [NEW] The computer-readable medium of claim 26 wherein said computer-readable instructions control said computer to receive user input which defines an action selection presentation pattern which defines hierarchically how users access said one or more service presentation patterns, one or more instance presentation patterns, one or more class population presentation patterns and one or more master-detail presentation patterns.

28. [NEW] The computer-readable medium of claim 27 wherein said computer-readable instructions control said computer to receive user input which defines said service presentation patterns, said instance presentation patterns, said class population presentation patterns and said master-detail presentation patterns by means of definition of other auxiliary primitives including but not limited to introduction patterns, defined selection patterns, dependency patterns, population selection patterns, display set patterns, filter patterns and order criterion patterns.

29. [NEW] The computer-readable medium of claim 25 wherein said computer-readable
instructions control said computer to receive user input data which defines classes within said object model each of which has services which may be events or transactions and include at least a creation event to create an instance of said class and create links of said instance to instances of classes related to said class, an optional destruction event which breaks links of a given instance of said class to instances of related classes and destroys said given instance of said class, and an optional set of modification events whose

functionality is defined in said functional model by means of valuations where a valuation is a primitive which creates a relationship between a variable attribute and an event of a class so as to define how the occurrence of said event affects the value of said variable attribute.

30. [NEW] The computer-readable medium of claim 25 wherein said computer-readable instructions control said computer to receive user input data which defines the functionality of transaction type services, which can be local or global, in terms of a sequence and/or alternation of other services, said services being events and/or transactions, with the functionality of transactions defined by a formula which defines which services the transaction encompasses and the value assigned to every argument of every service comprising said transaction.

31. [NEW] The computer-readable medium of claim 25 wherein said computer-readable instructions control said computer to receive user input data which defines secondary primitives which enrich the functionality of valuations of events and formulae of both local and global transactions by constraining said functionality using constraint primitives which include but are not limited to preconditions, integrity constraints, valid object lives, agent relationships and trigger relationships.

32. [NEW] The computer-readable medium of claim 31 wherein said computer-readable instructions control said computer to receive user input data which defines said preconditions as a condition which must be satisfied for a service to which said precondition is associated to execute, when executed by a given agent class, along with an error message to be displayed when said condition is not satisfied.

33. [NEW] The computer-readable medium of claim 31 wherein said computer-readable instructions control said computer to receive user input data which defines said integrity constraint as a condition which must be satisfied at any given instant by instances of a class so as to prevent services from changing the state of objects in a class to a non-desired state, along with an error message to be displayed when said condition is not satisfied.

34. [NEW] The computer-readable medium of claim 31 wherein said computer-readable instructions control said computer to receive user input data which defines valid lives for instances of a class by defining if a service specified by said user input can be executed by a given agent on a given instance of said class depending on the services that were previously executed on said given instance so as to constrain the execution of services on instances based on the history of services previously executed on said instances.

35. [NEW] The computer-readable medium of claim 31 wherein said computer-readable instructions control said computer to receive user input data which defines a relationship between two classes one playing the role of the "agent class" and the other playing the role of the "server class" so as to designate which attributes of said server class will be observable by said agent class and which services of said server class will be executable by said agent class so as to enable a user to enter input data which limits the accessibility of users who log onto the system as instances of said agent class to functionality of said target computer program such that some services can be blocked from

access by some users and which limits accessibility to the state of the system such that some attributes of a class or entire classes can be blocked from query by designated users.

36. [NEW] The computer-readable medium of claim 31 wherein said computer-readable instructions control said computer to receive user input data which defines trigger relationships to enrich the functionality of the system by specifying mandatory, and unnoticeable to the user, execution of a service whenever a Boolean condition holds, said condition specified on the state of an object.

37. [NEW] A computer-readable medium having stored thereon a data structure comprising:

a collection of fields that store data entered by a user defining a conceptual model of a computer program that the user wants automatically written by a translator process running on a computer, said conceptual model comprised of:

an object model data structure which defines a system class architecture comprised of one or more classes of objects defined by user input, each class having attributes the value of which collectively define the state of a system defined by said conceptual model (hereafter the target program);

a dynamic model data structure which defines valid object life cycles and which interobject communications can be established and defined by user input;

a functional model data structure which defines the semantics associated with any change of an object state as a consequence of an

event occurrence, user input defining data structures which define the class and which variable attribute of said class is affected by which event of that class and how the event affects the value of said variable attribute, said user input also defining a mathematical or logical valuation that defines how said variable attribute's value will be changed when said event happens;

a presentation model data structure defined by user input and which defines a full user interface defining how users or other processes will be able to interact with the functionality of the target program, said user input defining a set of patterns that specify which services will be available to user of the system and which information about the state of the target program users of the target program will be able to query.

38. The computer-readable medium of claim 37 wherein said data structures have user input data which define three types of events:

a creation event for each class which creates an instance of the class owning said creation event and provides values to all constant and variable attributes of the class which are required upon creation and establishing all relationships with instances of classes which are required to have a relationship with the instance of the class owning said creation event;

a destruction event for each class which eliminates all relationships between a given instance of the class owning said destruction event and other classes having relationships with the class owning said destruction event; or

a modification event for each class which modifies the values of one or more variable attributes of a given instance of the class owning said modification event, the effect of said modification being defined by one or more valuations linked

to said one or more variable attributes of said given instance of said class owning said modification event.

39. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer cause said computer to carry out the following process:

soliciting and receiving user input and converting said user input to data structures which define primitives of a conceptual model comprising an object model comprising one or more classes of objects which have attributes the state of which collective define the state of the system, and a dynamic model, and a functional model and a presentation model;

automatically converting said data structures of said conceptual model into statements according to the syntax and semantics of a formal language, said collection of formal language statements defining a formal language specification which defines the functionality of a computer program to be automatically written (hereafter the target program) by a computer executing a translator process, said target program being a full computer program having a user interface specified by data structures comprising said presentation model, and not a prototype which needs additional code, code components, code libraries or any third party software artifact and which will be the full functional equivalent of said conceptual model;

and wherein said data structures define the functionality of said conceptual model in terms of services which change the state of a system implemented by said target program, said services being events or local or global transactions, and wherein events are defined as the smallest execution units possible in the scope of a class within said object model in that it is not possible to decompose an event into

more elementary execution units, and wherein said local or global transactions are molecular execution units that have their functionality defined in terms of a sequence and/or alternation of other services each of which can be either an event or transaction, and wherein the functionality of a local or global transaction is explicitly defined by a data structure which defines a formula which expresses which services the transaction encompasses and the value assigned to every argument of every service comprising the transaction, and wherein said data structures which define local transactions limit the effect of a local transaction to affecting the state of primarily an instance of the class owning said transaction and potentially instances of classes related with the class owning said transaction, and wherein global transactions are not limited to affecting the state of only one instance of a class and can affect the state of any instance or set of instances of any class or set of classes;

and wherein said solicited user input includes user input which defines valuations for said functional model, where each valuation is a primitive relating a variable attribute and an event of a class and defines how the occurrence of said event affects the value of said variable attribute thereby explicitly defining the functionality of said event.

40. [NEW] The computer-readable medium of claim 39 wherein said computer-readable instructions, when executed by a computer, control said computer such that solicitation of user input defining said valuations is such that user input defining data structures which define different valuations for each variable attribute can be entered, each valuation relating said each variable attribute with the same or different events.

41. [NEW] The computer-readable medium of claim 39 wherein said computer-readable instructions, when executed by a computer, control said computer to solicit user input defining valuation data structures which can define valuations of different categories for each said variable attribute.

42. [NEW] The computer-readable medium of claim 39 wherein said computer-readable instructions, when executed by a computer, control said computer to solicit user input defining valuation data structures such that each variable attribute can include a list of valuation data structures that define how the variable attribute's value, and therefore the object's state, is changed by means of the different events.

43. [NEW] The computer-readable medium of claim 39 wherein said computer-readable instructions, when executed by a computer, control said computer to solicit user input defining valuation data structures such that each valuation data structure defines an optional condition that must be satisfied to apply the effect of the valuation on the value of the attribute, the event that will cause the valuation to be executed and the effect of the event to a specified attribute.

EVENT CLAIMS

44. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer cause said computer to perform the following process:

receive user input which defines a creation event an optional destruction event and an optional set of modification events for a class which is to become part of an object model of a conceptual model of a computer program to be automatically

written, said user input defining said creation event creating an instance of a class owning said creation event and providing a value for all constant and variable attributes of the class which are required upon creation, said user input also establishing all required relationships with instances of classes related with the instance of the class owning said creation event, and wherein said user input defines a destruction event which eliminates all relationships between a given instance of a class owning said destruction event and related instances of classes which have relationships with said class and then destroying the instance of the class owning said destruction event, and wherein said user input defines a set of modification events each of which uses user input to modify the values of one or more variable attributes of a given instance of the class owning said modification event, the effect of said modification defined by one or more logical or mathematical valuations defined by user input and which affect the value of one or more variable attributes upon the occurrence of said modification event;

converting said user input into data structures which define an object model having a class having said creation, destruction and modification events.

45. [NEW] The computer-readable medium of claim 44 wherein said computer-readable instructions stored thereon are such that they cause a computer executing them to solicit user input which defines said valuations as push-pop, state-independent or discrete domain.

46. [NEW] The computer-readable medium of claim 44 wherein said computer-readable instructions stored thereon are such that they cause a computer executing them to solicit user input which defines said creation, destruction and modification events and

said valuations, said solicitation by means of any suitable user interface mechanisms which prompt a user for the needed input to define said creation, destruction and modification events and said valuations.

VALUATION CLAIMS

47. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer cause said computer to perform the following process:

receiving user input to define a valuation which will affect the value of a variable attribute of a class upon the occurrence of an event and the satisfaction of an optional condition, said solicited user input also defining which variable attribute of which class will be affected and the event which will trigger said valuation to affect the value of said variable attribute; and

converting said user input into data structures which define said valuation as part of a conceptual model of a computer program to be automatically written.

48. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define the effect of said valuation with a formula referred to as a "valuation effect" using constants, values of input arguments of said event, values of other attributes of the class owning the variable attribute or an ancestor of it, or values of attributes of classes or ancestors of classes related to the class owning the variable attribute, and to convert said solicited user input into one or more data structures which define said valuation effect formula.

49. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said optional condition for the valuation with a Boolean formula referred to as a "valuation condition" that may be formed using constants, values of input arguments of said event, values of other attributes of the class owning the variable attribute or an ancestor of it, or values of attributes of classes or ancestors of classes related with the class owning the variable attribute, and converting said user input into data structures which define said Boolean valuation condition formula.

50. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said valuation such that it has both a valuation effect formula and a valuation condition formula, said solicited user input defining said valuation effect formula and said valuation condition formula by defining the valuation effect formula the value said variable attribute affected by said valuation will take upon occurrence of said event providing the valuation condition formula evaluates to a Boolean value of true, and to control said computer to use said solicited user input to form one or more data structures which define said valuation which has both said valuation effect and said valuation condition.

51. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said valuation such that it has only a valuation effect formula, and no valuation condition formula defining, by means of said valuation effect formula, the value said variable attribute will take upon occurrence of said event unconditionally, and controlling said computer to convert said solicited user input into one or more data

structures which implement said valuation in a conceptual model.

52. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said valuation such that it falls within one of the following three categories: push-pop, state-independent and discrete-domain, and for controlling said computer to convert said user input into one or more data structures which implement said valuation in the selected category.

53. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said valuation such that it falls within a push-pop category, said push-pop valuation being a valuation where said event which triggers said valuation increases or decreases the value of the variable attribute affected by said valuation by a given quantity or resets the value of the variable attribute affected by said valuation to a certain value, and controlling said computer to convert said solicited user input into one or more data structures which implements said push-pop valuation.

54. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said valuation such that it falls within a state-independent valuation category, wherein said state-independent valuation is one whose valuation effect formula provides a new value to the variable attribute affected by said valuation regardless of the value said variable attribute has at the time the event which triggers said valuation occurs, and said computer-readable instructions controlling said computer to convert said solicited user input

to create one or more data structures which implement said state-independent valuation.

55. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define said valuation such that it falls within a discrete domain category, wherein said discrete domain valuation is one whose valuation effect formula provides a new value for the variable attribute affected by said valuation depending upon the current value of said variable attribute, and said computer-readable instructions controlling said computer to convert said solicited user input into one or more data structures which implement said discrete domain valuation.

56. [NEW] The computer-readable medium of claim 47 wherein said computer-readable instructions stored thereon are such as to control said computer to solicit user input to define one or more said valuations each of which affects the value of the same variable attribute upon occurrence of the same event, regardless of the category of said valuation, if and only if one of said valuations and only one of them has only a valuation effect formula but no valuation condition formula, and the rest of them compulsorily have both a valuation effect formula and a valuation condition formula, said computer-readable instructions for converting said solicited user input data into one or more data structures which implement said one or more valuations.

Transaction Claims

57. [NEW] A computer-readable medium having computer-readable instructions stored thereon, which, when executed by a computer, control said computer to solicit user

input to define a transaction wherein said transaction is a molecular execution unit expressed in terms of a formula that specifies services in the form of events or transactions which together comprise said molecular execution unit, and said computer-readable instructions controlling said computer to convert said user input into one or more data structures which implement said transaction.

58. [NEW] The computer-readable medium of claim 57 wherein said computer-readable instructions control said computer to solicit user input which defines said transaction in terms of a sequence and/or alternation of services, and said computer-readable instructions controlling said computer to convert said user input into one or more data structures which implement said transaction.

59. [NEW] The computer-readable medium of claim 57 wherein said computer-readable instructions control said computer to solicit user input which defines said transaction in terms of an "all-or-nothing" execution policy in the sense that the execution of every service comprising said transaction must be successful in creating changes to the state of the system for the execution of said transaction to be successful , and the failure in execution of any service comprising said transaction in creating changes to the state of the system means failure in the execution of said transaction and causing to override changes to be made to the state of the system in that the changes made to the state of the system by any of the services comprising said transaction will be reversed, said computer-readable instructions controlling said computer to convert said user input into one or more data structures implementing said "all-or-nothing" transaction.

60. [NEW] The computer-readable medium of claim 57 wherein said computer-

readable instructions control said computer to solicit user input which defines said transaction as a local transaction wherein said one or more services which comprise said local transaction are events or transactions owned by the class or the ancestor of the class which owns said transaction which owns said transaction formula, or said one or more services which comprise said transaction are owned by classes related with the class owning the transaction which owns said transaction formula, and wherein said computer-readable instructions controlling said computer to convert said user input into one or more data structures which implement said local transaction.

61. [NEW] The computer-readable medium of claim 57 wherein said computer-readable instructions control said computer to solicit user input which defines said transaction as a global transaction wherein said one or more services which comprise said global transaction are events and/or local transactions owned by any of the class in a conceptual model and/or global transactions.

62. [NEW] The computer-readable medium of claim 57 wherein said computer-readable instructions control said computer to solicit user input which defines said transaction as a local transaction, wherein said one or more services which comprise said local transaction have arguments whose value is determined in said transaction formula by contained formulas formed by constants, values of input arguments of said transaction, values of attributes of the class owning said transaction or an ancestor of it, or values of attributes of classes or ancestors of classes related with the class owning said transaction, and said computer-readable instructions controlling said computer to convert said user input into one or more data structures which implement said local transaction.

63. [NEW] The computer-readable medium of claim 57 wherein said computer-readable instructions control said computer to solicit user input which defines said transaction as a global transaction, wherein said one or more services which comprise said global transaction have arguments whose value is determined in said transaction formula by contained formulas formed by constants, values of input arguments of said transaction, or values of attributes of any class in said conceptual model, said computer-readable instructions also being such as to control said computer to convert said user input into one or more data structures which implement said global transaction.

CLAIMS ABOUT SPECIFYING INTERFACE MECHANISMS TO INTERFACE TO FUNCTIONALITY OF THE SYSTEM

64. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer control said computer to perform the following process:

soliciting user input to define primitives of a conceptual model defining the functionality of a target computer program to be written automatically, said conceptual model comprised of an object model, a dynamic model, a functional model and a presentation model;

converting said user input into data structures which implement said conceptual model in the form of statements in a formal language syntax that together comprise a formal language specification, said formal language specification defining a complete computer program having a user interface and which needs no additional code, code components, code libraries or any other third party software artifact in order to be functionally equivalent to the conceptual model;

and wherein said data structures and formal language specification define the functionality of the conceptual model in terms of how the state of the system is changed by means of services, said services being events and transactions where transactions are comprised of other services and falling in either a local or global category;

and wherein said data structures and formal language specifications define said presentation model in the form of user interface mechanisms which define how a user or other processes will be able to interact with the functionality of the system.

65. [NEW] The computer-readable medium of claim 64 wherein said computer-readable instructions stored thereon are such as to solicit user input which defines said presentation model by specifying a set of patterns that specify the interaction means through which services will be available to user of the system and through which information about the state of the system users of the system will be able to query.

66. [NEW] The computer readable medium of claim 64 wherein said computer-readable instructions stored thereon are such as to solicit user input which defines said presentation model in terms of a set of service presentation patterns, instance presentation patterns, class population presentation patterns, master/detail presentation patterns and an action selection presentation pattern.

67. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer control said computer to perform the following process:

soliciting user input to define primitives of a presentation model which is part of a conceptual model of a target computer program to be automatically written, said

presentation model defining a user interface by a set of patterns which will be used to define how users can interact with said target computer program by specifying the interaction scenarios for services which will be available to users of the target computer program and the interaction scenarios to query information about the state of the target computer program which users of the system will be able to query;

and converting said user input to one or more data structures which define said patterns of said presentation model.

68. [NEW] The computer readable medium of claim 67 wherein said computer-readable instructions stored thereon are such as to solicit user input which defines said presentation model in terms of a set of service presentation patterns, instance presentation patterns, class population presentation patterns, master/detail presentation patterns and an action selection presentation pattern.

69. [NEW] The computer readable medium of claim 68 wherein said computer-readable instructions stored thereon are such as to solicit user input which defines said presentation model in terms of an action selection presentation pattern which specifies in a hierarchical way what service presentation patterns, instance presentation patterns, class population presentation patterns and master-detail presentation patterns will be offered by the system as a means for users of said target computer program to interact with it.

70. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer, cause said computer to carry out the following process:

solicit user input which defines a presentation model which defines the user

interface for a target computer program defined by a conceptual model, said solicited user input defining at least a service presentation pattern for every service of the system specifying how user of the system will be able to invoke a service of said system and said solicited user input further defining:

an optional introduction pattern assigned to every input data valuated argument of a service to which said service presentation pattern is assigned;

an optional defined selection pattern assigned to every input data valuated argument of the service to which said service presentation pattern is assigned;

an optional population selection pattern assigned to every input object valuated argument of a service to which said service presentation pattern is assigned;

an optional dependency pattern assigned to every input argument of a service to which said service presentation pattern is assigned; and

converting said solicited user input into one or more data structures which implement said service presentation pattern.

71. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer, cause said computer to carry out the following process:

solicit user input which defines a presentation model which defines the user interface for a target computer program defined by a conceptual model, said solicited user input defining at least an instance presentation pattern for every class of the system specifying how users of said target computer program will be able to query the state of a given instance of a class of said target computer program, said solicited user input further

comprising user input which defines:

a display set pattern;

a set of available services of the class owning said instance presentation pattern;

a set of navigations to presentation patterns owned by the classes related to a class owning said instance presentation pattern; and

converting said user input into one or more data structures implementing said instance presentation pattern.

72. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer, cause said computer to carry out the following process:

solicit user input which defines a presentation model which defines the user interface for a target computer program defined by a conceptual model, said solicited user input defining at least a class population presentation pattern for every class of the system specifying how users of said target computer program will be able to query the population of a given class of said target computer program, said solicited user input further comprising user input which defines:

a display set pattern;

an optional filter pattern;

an optional order criterion pattern;

a set of available services of the class owning said class population presentation pattern;

a set of navigations to presentation patterns owned by the classes related to a class owning said class population presentation pattern; and

converting said user input into one or more data structures implementing said class population presentation pattern.

73. [NEW] A computer-readable medium having stored thereon computer-readable instructions which, when executed by a computer, cause said computer to carry out the following process:

solicit user input which defines a presentation model which defines the user interface for a target computer program defined by a conceptual model, said solicited user input defining at least a master/detail presentation pattern for every class of the system specifying how users of said target computer program will be able to query the state of a given instance of a class of said target computer program owning said master/detail presentation pattern (or the population of a class of said target computer program owning said master/detail presentation pattern), and the population of instances of classes related with an instance of the class of said target computer program owning said master/detail presentation pattern with said solicited user input further comprising user input which defines:

an instance presentation pattern, or a class population presentation pattern, defined for the class owning said master/detail presentation pattern, referred to as "master presentation pattern";

a set of presentation patterns which may be of type instance presentation pattern, class population presentation pattern, or master/detail presentation pattern, defined for classes related with the class owning the master/detail presentation pattern, referred to as "detail presentation patterns"; and

converting said user input into one or more data structures implementing said instance presentation pattern.